COURSE UNIT (MODULE) DESCRIPTION

Course unit (module) title	Code
Systemic programming	

Academic staff	Core academic unit(s)
Coordinating: dr. Liudvikas Kaklauskas **Other:**	Šiaulių akademija

Study cycle	Type of the course unit
Bachelor studies	Mandatory

Mode of delivery	Semester or period when it is delivered	Language of instruction
Auditorium	5 semester	English

Requirements for the student	
Prerequisites: Fundamentals of programming, structured programming, object-oriented programming	**Co-requisites (if relevant):**

Number of ECTS credits allocated	Student's workload (total)	Contact hours	Individual work
5	133	48	85

Purpose of the course unit		
To provide knowledge and learn how to apply systematic programming methods to create programs		
Learning outcomes of the course unit	**Teaching and learning methods**	**Assessment methods**
BK1.4. Apply knowledge of program systems, creating secure and other relevant criteria-compliant informatics applied solutions to solve relevant problems of professional activity.	Theoretical lecture, laboratory work, defense of laboratory work, search of scientific literature.	Defense of laboratory work, test (exam).
BK3.3. Demonstrate creativity in solving tasks and problems of professional activity.	Theoretical lecture, laboratory work, defense of laboratory work.	Defense of laboratory work, test (exam).
DK2.3. To analyze the data, information and solutions needed to solve the actual problem of the professional activity of program systems using effective methods according to various criteria.	Theoretical lecture, laboratory work, defense of laboratory work.	Defense of laboratory work, test (exam).
DK3.6To implement a program system product or service to solve the actual problem of professional activity according to the functional and non-functional requirements for the program system.	Theoretical lecture, laboratory work, defense of laboratory work, remote consultations.	Defense of laboratory work, test (exam).

Content	Contact hours							Individual work: time and assignments	
	Lectures	Tutorials	Seminars	Workshops	Laboratory work	Internship	**Contact hours, total**	**Individual work**	**Tasks for individual work**
1. Analysis of microprocessors and computer resources.	2				2		**4**	**6**	Analysis of scientific literature on computer structure and its management
2. Software model of processor architecture.	2				2		**4**	**9**	Preparation and defense of laboratory works.
3. Writing the program with assembler, data description.	2				2		**4**	**8**	Preparation and defense of laboratory works.
4. Management of macro processors, loaders, compilers.	2				2		**4**	**9**	Preparation and defense of laboratory works.
5. Use of operating system resources.	2				3		**5**	**9**	Preparation and defense of laboratory works.
6. Programmatic management of files and processes.	2				3		**5**	**10**	Preparation and defense of laboratory works.
7. PC hardware management.	3				2		**5**	**9**	Preparation and defense of laboratory works.
8. Development of the simplest program for the operating system of 32 and 64-bit architecture.	3				2		**5**	**8**	Preparation and defense of laboratory works.
9. Modular program, implementation of data exchange.	3				3		**6**	**10**	Preparation and defense of laboratory works.
10. Implementation of interaction between assembler and high-level programming languages.	3				3		**6**	**9**	Preparation and defense of laboratory works.
Total:	24				24		**48**	**85**	

Assessment strategy	Weight %	Deadline	Assessment criteria
Defense of laboratory work (G)	50%	At a fixed time during the semester	The completed laboratory works and their defense are evaluated (the evaluations of each laboratory work and its defense are averaged and multiplied by a weighting factor of 5%, a total of 10 laboratory works)
Exam (E)	50%	At a fixed time during the session	A test consisting of closed and open type questions is held

Author (-s)	Publishing year	Title	Issue of a periodical or volume of a publication	Publishing house or web link
Required reading				
Johnson M. Hart.	2010	Windows System Programming, fourth edition.		Pearson Education
Love R.	2013	LINUX system programming.		O’Reilly Media
Anthony R. J.	2015	Systems Programming Designing and Developing Distributed Applications.		Elsevier Inc.
Recommended reading				
Sanchez J., M/P/Canton.	2015	Microcontrollers HIGH-		CRC Press.

		PERFORMANCE SYSTEMS AND PROGRAMMING.		
Tanenbaum A.S.	2007	Operating systems. Design and implementation.		New Jersey, 2007.
Kerrisk M.	2010	The Linux programming interface.		No Starch Press, Inc.